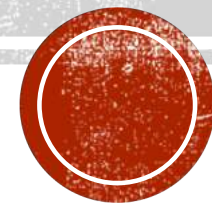


# LABORATÓRIO 1

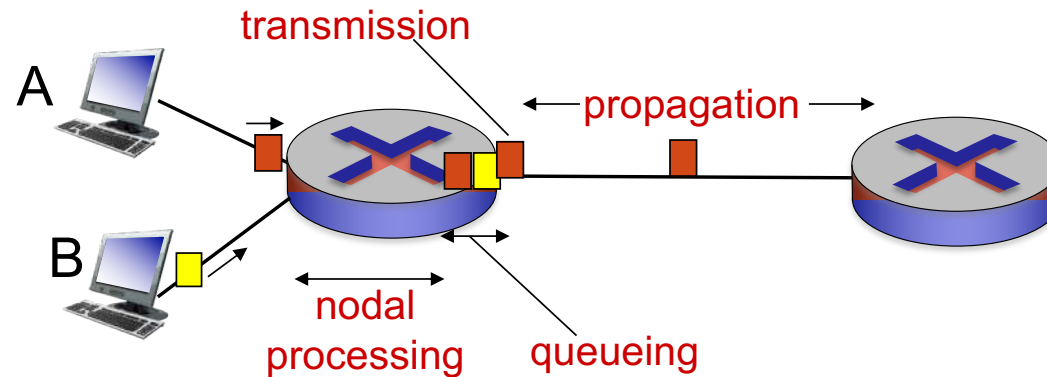


Cesar Marcondes – [cmarcondes@ita.br](mailto:cmarcondes@ita.br)

Instituto Tecnológico da Aeronáutica (ITA)



# FONTES DE ATRASO DE REDE



$d_{proc}$ : processamento nodal

- checagem erros bits
- determinar saída
- tipicamente < msec

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

$d_{queue}$ : atraso de fila

- tempo de espera até a transmissão de saída
- depende do nível de congestionamento

$d_{trans}$ : atraso de transmissão:

- $L$ : tamanho do pacote (bits)
- $R$ : banda do enlace (bps)
- $d_{trans} = L/R$

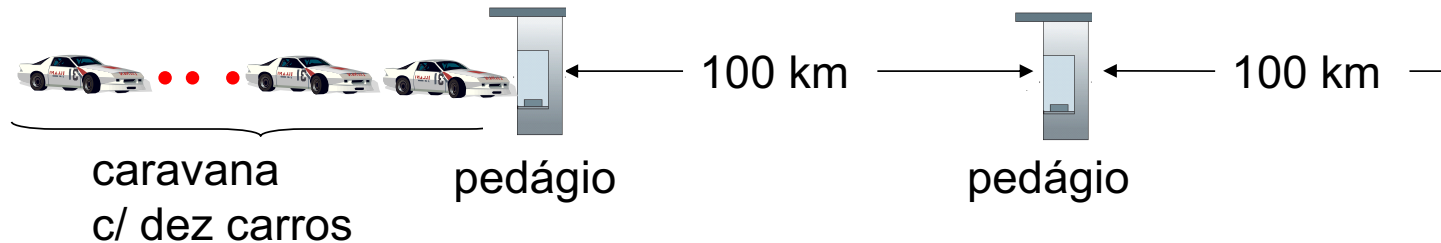
$d_{prop}$ : atraso de propagação:

- $d$ : distância geográfica do enlace físico
- $s$ : tempo de propagação dos eletrons ou luz no enlace (ex. vel. da luz  $\sim 2 \times 10^8$  m/sec)

$d_{trans}$  e  $d_{prop}$   
bem diferentes

$$d_{prop} = d/s$$

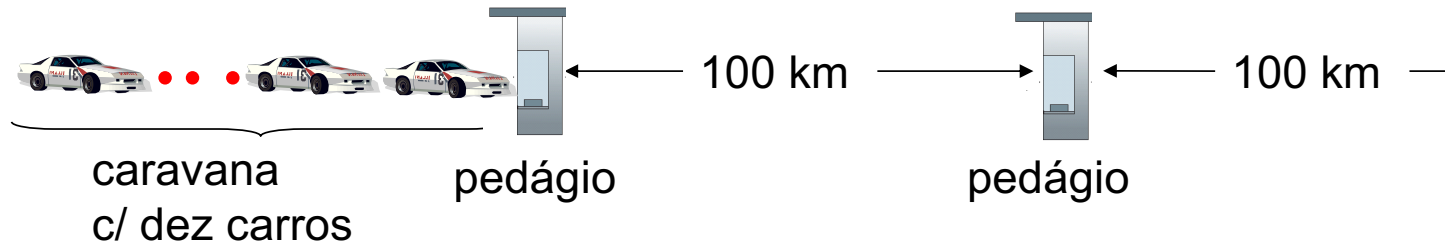
# ANALOGIA CARAVANA



- carros se “propagam” a 100 km/hr
- Praças de pedágio levam 12 segundos para **servir** o carro (equivalente tempo de transmissão do bit)
- carro ~ bit; caravana ~ pacote
- **Q: Quanto tempo para a caravana ficar alinhada na entrada do Segundo pedágio?**
- tempo para “passar” a caravana inteira através do primeiro pedágio e entrar na próxima estrada =  $12 \times 10 = 120$  sec
- Tempo para o ultimo carro se propagar do primeiro p/ segundo pedágio:  
 $100\text{km}/(100\text{km/hr}) = 1$  hr
- **R: 62 minutos**

**Servir** = coletar o pagamento do pedágio, aguardar o portão abrir, etc

# ANALOGIA CARAVANA



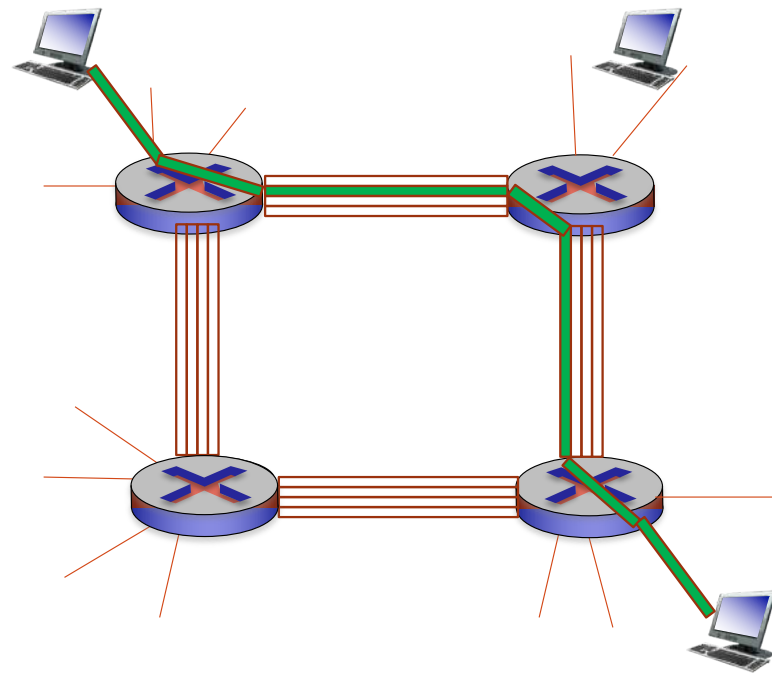
- vamos supor que os carros agora tenham velocidade de **1000 km/hr**
- e suponha que as praças de pedágio agora levem 1 minute (60 seg) para servir um carro
- **Q:** Voce acha que os carros chegarão no segundo pedágio antes de todos os carros da caravana saírem no primeiro pedágio?
- **R: Sim!** após 7 min, o primeiro carro chegará no Segundo pedágio; e ainda assim, três carros ainda estão aguardando no primeiro pedágio

# DE COMUTAÇÃO DE CIRCUITOS PARA COMUTAÇÃO DE PACOTES

- Alexander Graham Bell
  - Telefone (conexão par a par)
- Sistema de Comutação Manual



- Sistema de Comutação Eletrônica
- Multiplexação Digital

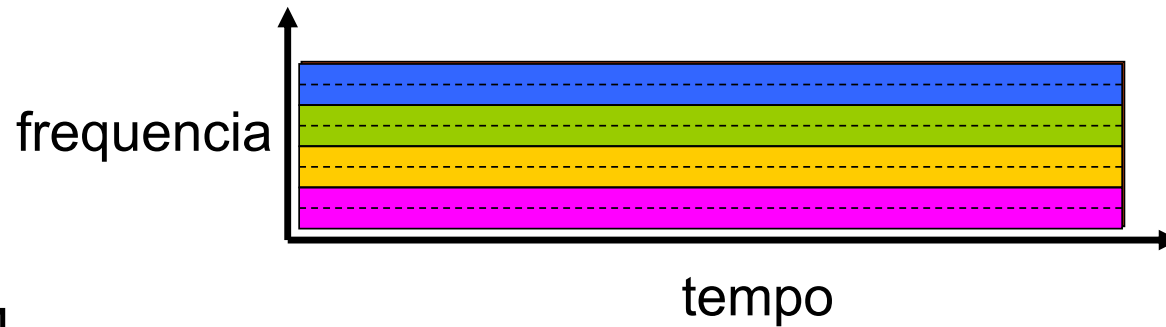


# COMUTAÇÃO DE CIRCUITOS: FDM VERSUS TDM

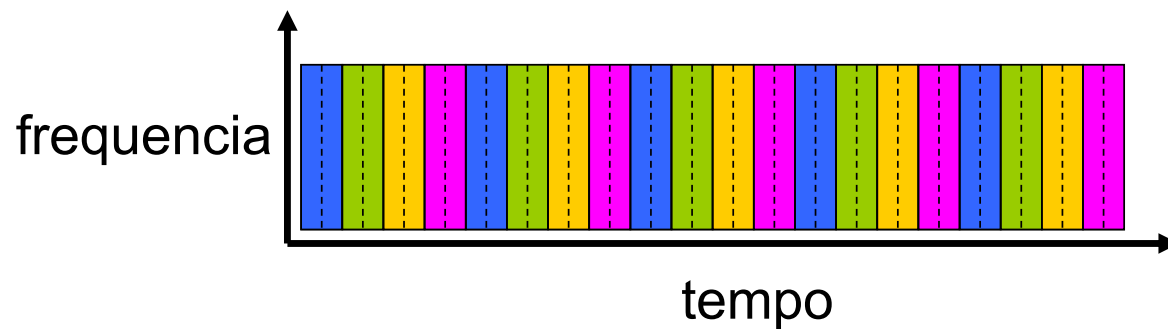
FDM

Exemplo:

4 usuários 



TDM

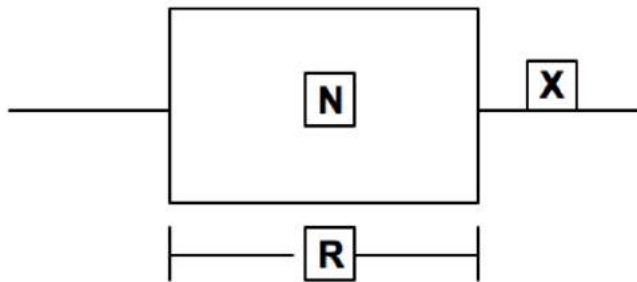


# MULTIPLEXAÇÃO ESTATÍSTICA

- Baseado em Teoria de Filas (Leonard Kleinrock - UCLA)
- Melhora a utilização da rede
- Permite flexibilidade de recuperação em caso de falhas
- Base da Internet
- Pressupõe filas (buffers) em todos os equipamentos



# POTPOURRI EM TEORIA DE FILAS



O número médio de clientes de um serviço em (dentro) uma "caixa preta" é igual a média do tempo que cada cliente gasta na "caixa" vezes a vazão da "caixa"

Lei de Little  $N = R * X$

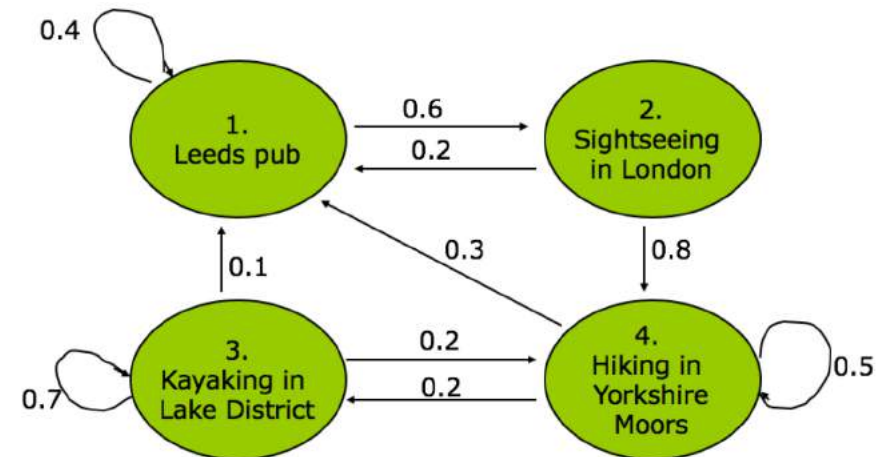
- Um grande serviço de portal oferece serviço de e-mail gratuito. O número de usuários registrados é de **2 Milhões** e **30% deles** enviam e-mails pelo portal durante o horário de pico. Cada **email leva em média 5 segundos** para ser processado e entregue na caixa de correio de destino. Durante o período ocupado, **cada usuário envia 3,5 mensagens** de email em média. O arquivo de log indica que o **tamanho médio de uma mensagem de email é de 7,120 bytes**.
- Qual deve ser a capacidade do "spool" para emails enviados durante o período de pico?





# POTPOURRI EM TEORIA DE FILAS

- Jovem passando um ano na Inglaterra; faz check-in com a mãe às 3 da tarde diariamente
- Modelo Markoviano: Enumeração do espaço de estados, Identificação de transição de estado, Parametrização
- Se em Leeds um dia, então no dia seguinte:
  - Visitas guiadas em Londres(60%)
  - Novamente no pub de Leeds (40%)
- Se em Londres um dia, então no dia seguinte:
  - ...
- O pai deseja saber qual é a percentagem de dias que o filho NÃO está bebendo em Leeds?
  - $1 - P1 = 1 - 0.2644 = 0.74$



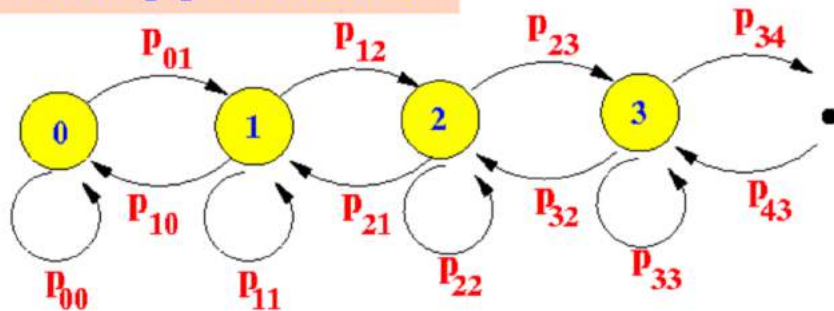
(Modelos Mais Sofisticados)  
Modelo Markoviano



# POTPOURRI EM TEORIA DE FILAS

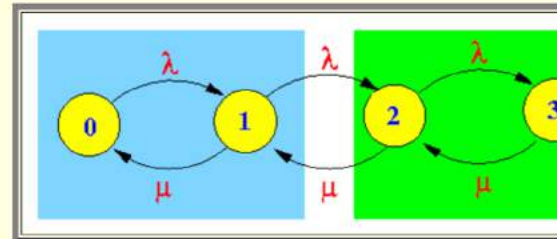
- Processo Markoviano de Nascimento e Morte

state  $k$  = population size is  $k$



- Fila M/M/1

Cut 2:



Final set of equations:

$$\begin{aligned} p_0 \lambda &= p_1 \mu \\ p_1 \lambda &= p_2 \mu \\ p_2 \lambda &= p_3 \mu \\ p_3 \lambda &= p_4 \mu \\ &\dots \end{aligned}$$

Or:

$$\begin{aligned} p_1 &= \lambda/\mu \times p_0 \\ p_2 &= \lambda/\mu \times p_1 \\ p_3 &= \lambda/\mu \times p_2 \\ p_4 &= \lambda/\mu \times p_3 \\ &\dots \end{aligned}$$

Or:

$$\begin{aligned} p_1 &= \lambda/\mu \times p_0 && \dots (1) \\ p_2 &= (\lambda/\mu)^2 \times p_0 && \dots (2) \\ p_3 &= (\lambda/\mu)^3 \times p_0 && \dots (3) \\ p_4 &= (\lambda/\mu)^4 \times p_0 && \dots (4) \end{aligned}$$

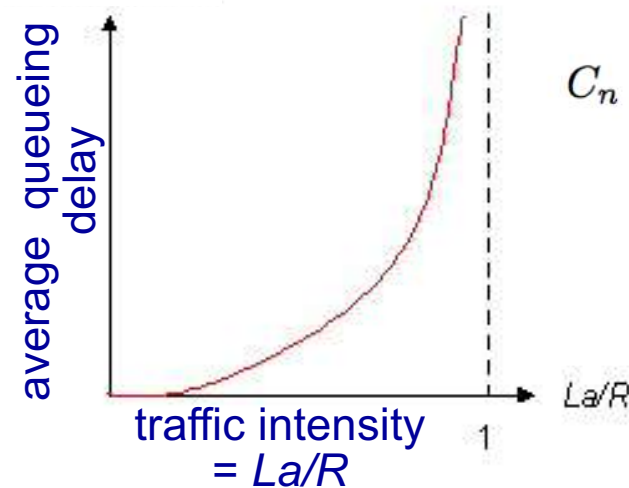
Equilibrium equation:

$$\begin{aligned} \text{Flow from left to right: } & \lambda \times p_1 \\ \text{Flow from right to left: } & \mu \times p_2 \\ \text{Equilibrium: } & \lambda \times p_1 = \mu \times p_2 \end{aligned}$$

# ATRASO DE ENFILEIRAMENTO

$$\rho = \frac{\lambda}{\mu} < 1.$$

- $R$ : banda do enlace (bps)
- $L$ : tamanho pacote (bits)
- $a$ : taxa de chegada de pacotes



$$C_n = \rho^n, \quad n = 0, 1, 2, \dots$$

$$P_0 = 1 - \rho,$$

$$L = \frac{\rho}{1 - \rho}.$$

- $La/R \sim 0$ : atraso de fila pequeno
- $La/R \rightarrow 1$ : atraso de fila grande
- $La/R > 1$ : mais “trabalho” chegando do que sendo servido, media do atraso aproxima do infinito (se o buffer infinito)!



$La/R \sim 0$

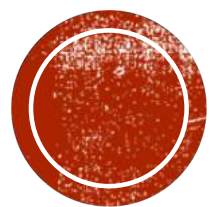


$La/R \rightarrow 1$

# EXPERIMENTOS COM APPLET'S KUROSE

- Abrir os seguintes links:
- [https://media.pearsoncmg.com/aw/ecs\\_kurose\\_compnetwork\\_7/cw/content/interactiveanimations/transmission-vs-propagation-delay/transmission-propagation-delay-ch1/index.html](https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/transmission-vs-propagation-delay/transmission-propagation-delay-ch1/index.html)
- [https://media.pearsoncmg.com/aw/ecs\\_kurose\\_compnetwork\\_7/cw/content/interactiveanimations/queueing-loss-applet/index.html](https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/queueing-loss-applet/index.html)
- [https://media.pearsoncmg.com/aw/ecs\\_kurose\\_compnetwork\\_7/cw/content/interactiveanimations/message-segmentation/index.html](https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/message-segmentation/index.html)
  
- Fazer os seguintes experimentos (em sala ou em casa):
- A) Experimentar com parâmetros de propagação e transmissão similares à Mars Laboratory – Opportunity - <https://mars.nasa.gov/msl/mission/communicationwithearth/data/>. Calcular quanto tempo para enviar uma foto de gelo em Marte de 4 MB.
- B) Calcular o tempo que leva para encher o buffer em condições de congestionamento na Applet. E ao acelerar de novo a transmissão, qual o tempo para esvaziar o buffer?
- C) Experimentar com diferentes tipos de segmentação e tamanhos e descrever a diferença dos mesmos.





**MÃOS A OBRA ...**

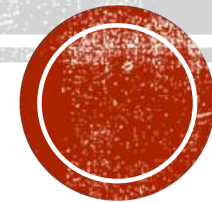


# LABORATÓRIO 1



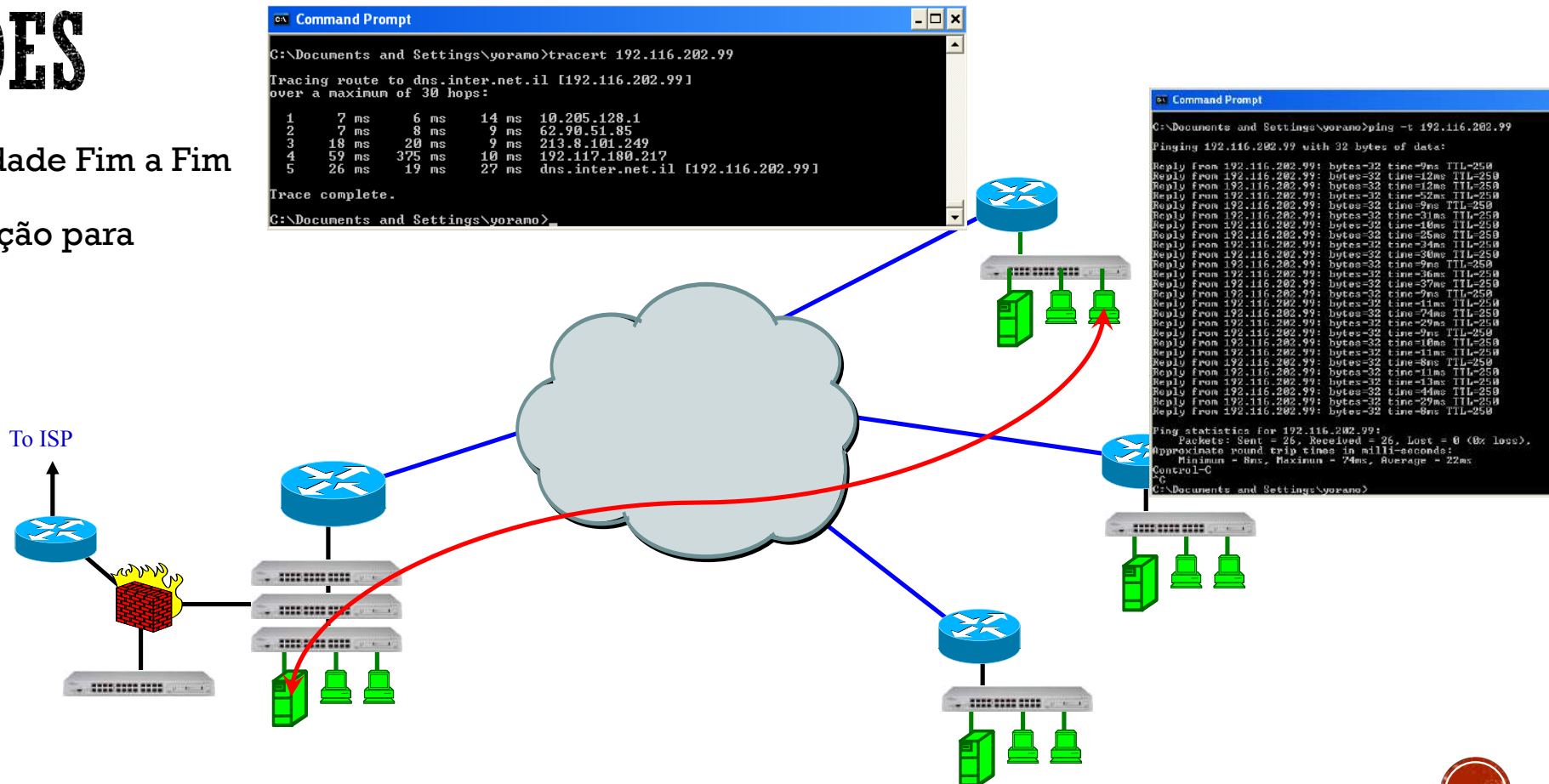
Cesar Marcondes – [cmarcondes@ita.br](mailto:cmarcondes@ita.br)

Instituto Tecnológico da Aeronáutica (ITA)



# FERRAMENTAS DE MONITORAÇÃO DE REDES

- Conectividade Fim a Fim
- Primeira ação para monitorar







# SNIFFERS (WIRESHARK)

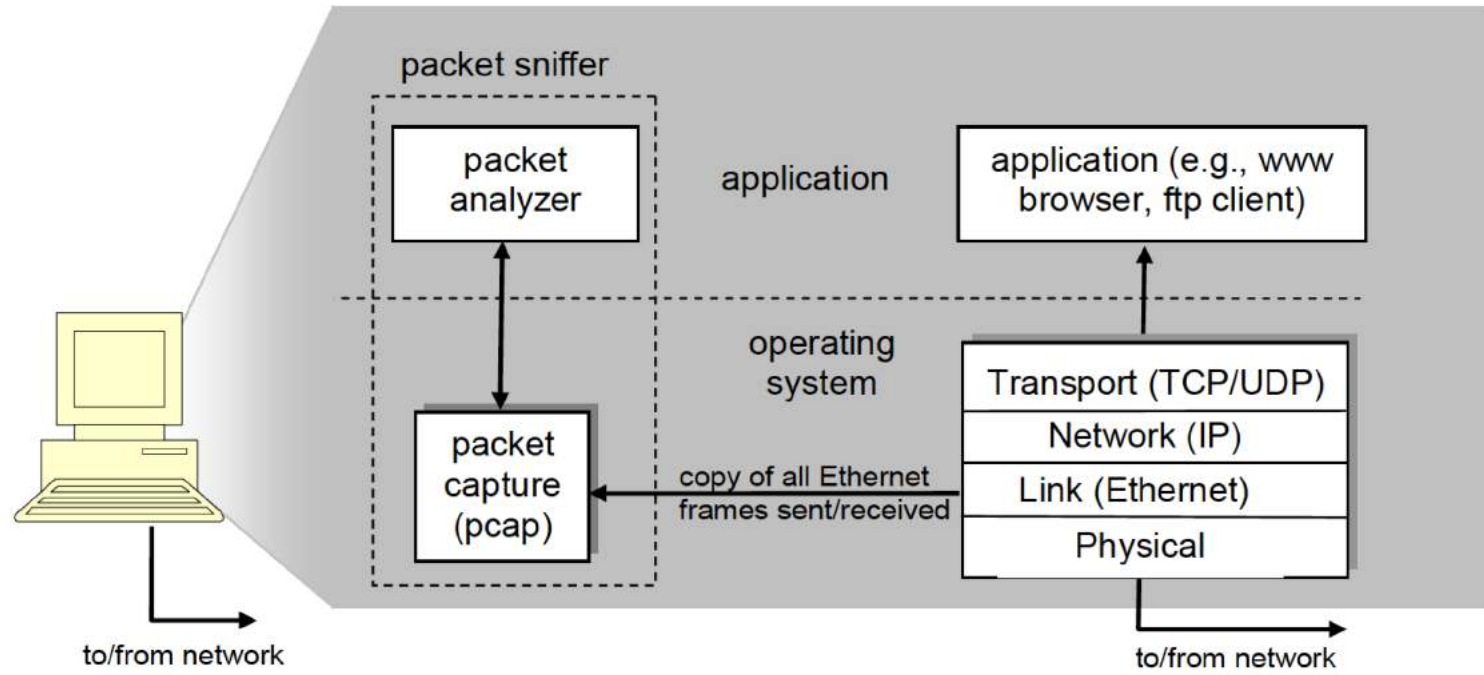


Figure 1: Estrutura de um sniffer de pacotes



# WIRESHARK

Capturing from Wi-Fi: en0

Apply a display filter... <36/> Expression...

| No. | Time      | Source                    | Destination     | Protocol | Length | Info  |
|-----|-----------|---------------------------|-----------------|----------|--------|---|
| 31  | 7.552628  | 192.168.15.20             | 192.168.15.148  | TCP      | 183    | 50273 → 8009 [PSH, ACK] Seq=118 Ack=118 Win=4096  |
| 32  | 7.556784  | 192.168.15.148            | 192.168.15.20   | TCP      | 183    | 8009 → 50273 [PSH, ACK] Seq=118 Ack=235 Win=444 L |
| 33  | 7.556795  | 192.168.15.20             | 192.168.15.148  | TCP      | 66     | 50273 → 8009 [ACK] Seq=235 Ack=235 Win=4092 Len=0 |
| 34  | 7.692450  | 162.125.34.129            | 192.168.15.20   | TCP      | 66     | 443 → 50353 [ACK] Seq=308 Ack=952 Win=360 Len=0 T |
| 35  | 8.164947  | 192.168.15.20             | 224.0.0.251     | MDNS     | 117    | Standard query 0x0000 PTR _afpovertcp._tcp.local, |
| 36  | 8.165874  | fe80::18e3:6dbe:8d35:e87d | ff02::fb        | MDNS     | 137    | Standard query 0x0000 PTR _afpovertcp._tcp.local, |
| 37  | 8.166728  | SamsungE_9b:6f:0c         | Broadcast       | ARP      | 60     | Who has 192.168.15.17 Tell 192.168.15.27          |
| 38  | 8.178863  | 192.168.15.27             | 224.0.0.7       | UDP      | 226    | 8001 → 8001 Len=184                               |
| 39  | 8.487600  | 192.168.15.20             | 224.0.0.251     | MDNS     | 117    | Standard query 0x0000 PTR _afpovertcp._tcp.local, |
| 40  | 8.488617  | fe80::18e3:6dbe:8d35:e87d | ff02::fb        | MDNS     | 137    | Standard query 0x0000 PTR _afpovertcp._tcp.local, |
| 41  | 8.651920  | 192.168.15.27             | 192.168.15.255  | UDP      | 77     | 46676 → 15600 Len=35                              |
| 42  | 9.490440  | 192.168.15.20             | 224.0.0.251     | MDNS     | 117    | Standard query 0x0000 PTR _afpovertcp._tcp.local, |
| 43  | 9.491421  | fe80::18e3:6dbe:8d35:e87d | ff02::fb        | MDNS     | 137    | Standard query 0x0000 PTR _afpovertcp._tcp.local, |
| 44  | 10.257312 | SamsungE_9b:6f:0c         | Broadcast       | ARP      | 60     | Who has 192.168.15.17 Tell 192.168.15.27          |
| 45  | 10.462340 | 192.168.15.27             | 224.0.0.7       | UDP      | 226    | 8001 → 8001 Len=184                               |
| 46  | 11.486174 | Wisol_96:04:6d            | Broadcast       | ARP      | 60     | Who has 192.168.15.17 Tell 192.168.15.56          |
| 47  | 11.691661 | 192.168.15.27             | 239.255.255.250 | UDP      | 77     | 49625 → 15600 Len=35                              |
| 48  | 12.305346 | SamsungE_9b:6f:0c         | Broadcast       | ARP      | 60     | Who has 192.168.15.17 Tell 192.168.15.27          |

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
 Ethernet II, Src: SamsungE\_9b:6f:0c (24:4b:03:9b:6f:0c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
 Address Resolution Protocol (request)

```

0000  ff ff ff ff ff ff 24 4b 03 9b 6f 0c 08 06 00 01  ....$K..o...
0010  08 00 06 04 00 01 24 4b 03 9b 6f 0c c0 a8 0f 1b  ....$K..o...
0020  00 00 00 00 00 00 c0 a8 0f 01 00 00 00 00 00 00  ....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ....
  
```

Wi-Fi: en0: <live capture in progress> Packets: 48 / Displayed: 48 (100.0%) Profile: Default

Wireshark - Go Deep.

Secure | <https://www.wireshark.org>

WIRESHARK NEWS Get Acquainted Get Help Develop Our Sponsor SharkFest

SharkFest '18 EUROPE is on! Find out more at <https://sharkfesteurope.wireshark.org>.

Download Get Started Now

Learn Knowledge is Power

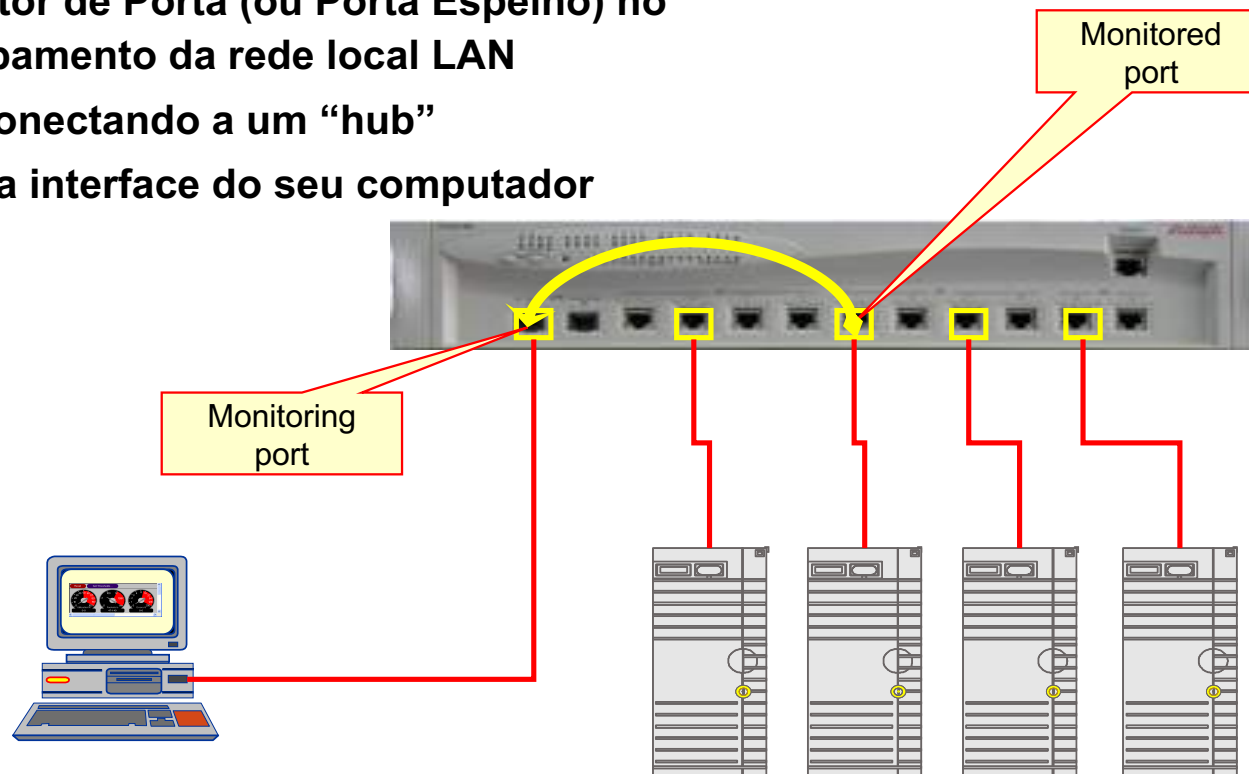
Go Beyond With Riverbed Technology



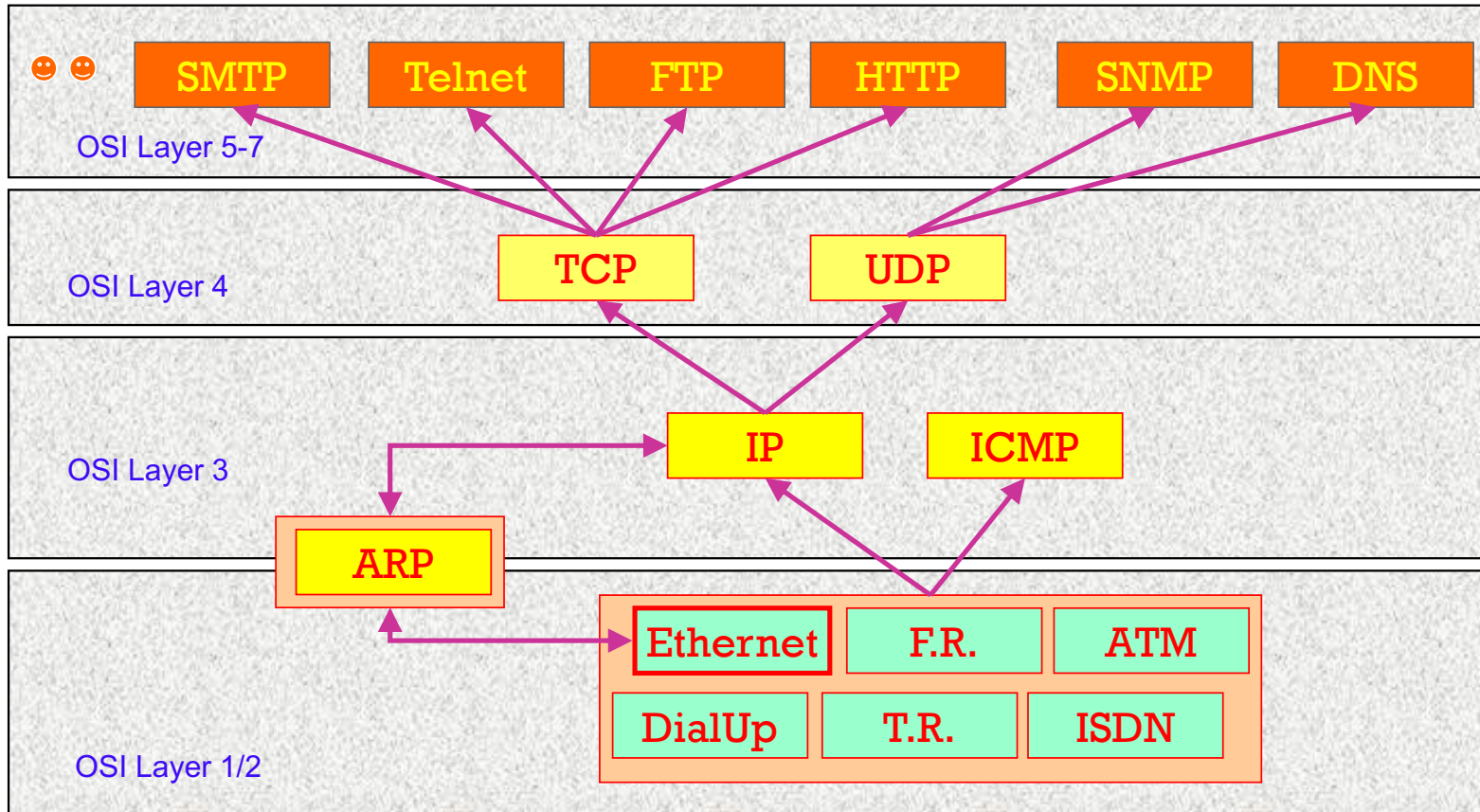
# COMO MEDIR USANDO WIRESHARK?

## \* Método de medição:

- Monitor de Porta (ou Porta Espelho) no equipamento da rede local LAN
- Ou conectando a um “hub”
- Ou da interface do seu computador

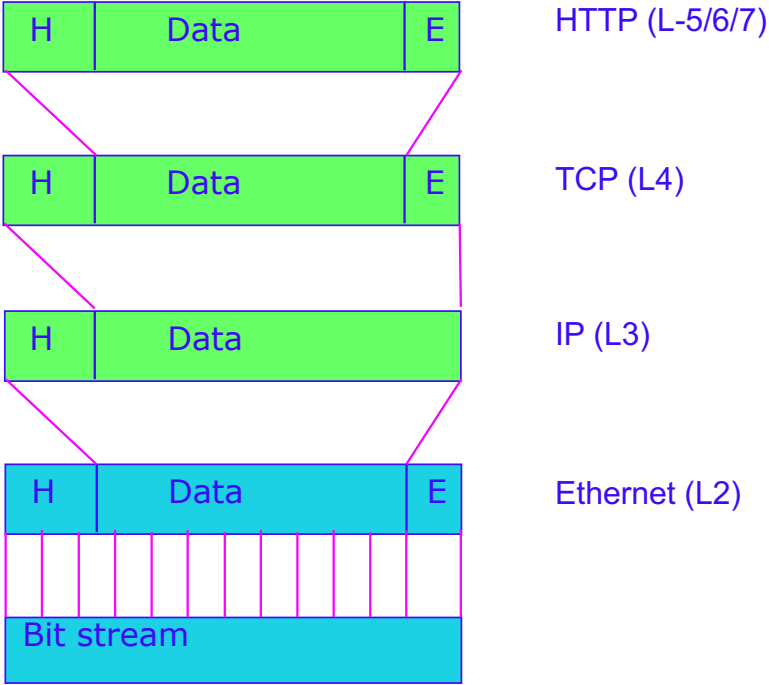


# PILHA DE PROTOCOLOS TCP/IP

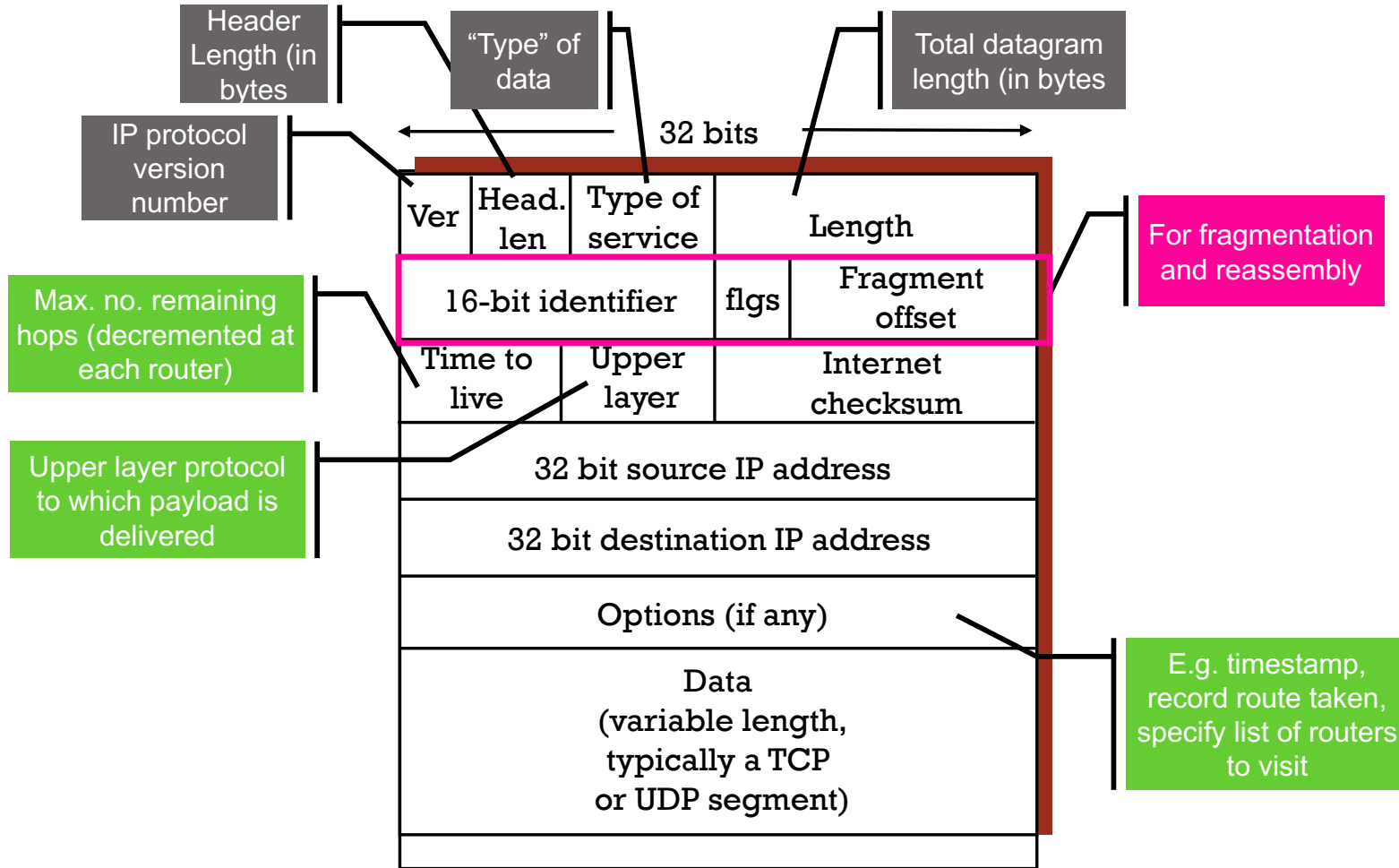


# ENCAPSULAMENTO DE CABEÇALHOS

This is the IP header



# FORMATO DATAGRAMA IP (V4)



# EXEMPLO WIRESHARK PACOTE IP

Filter:  Expression... Clear Apply

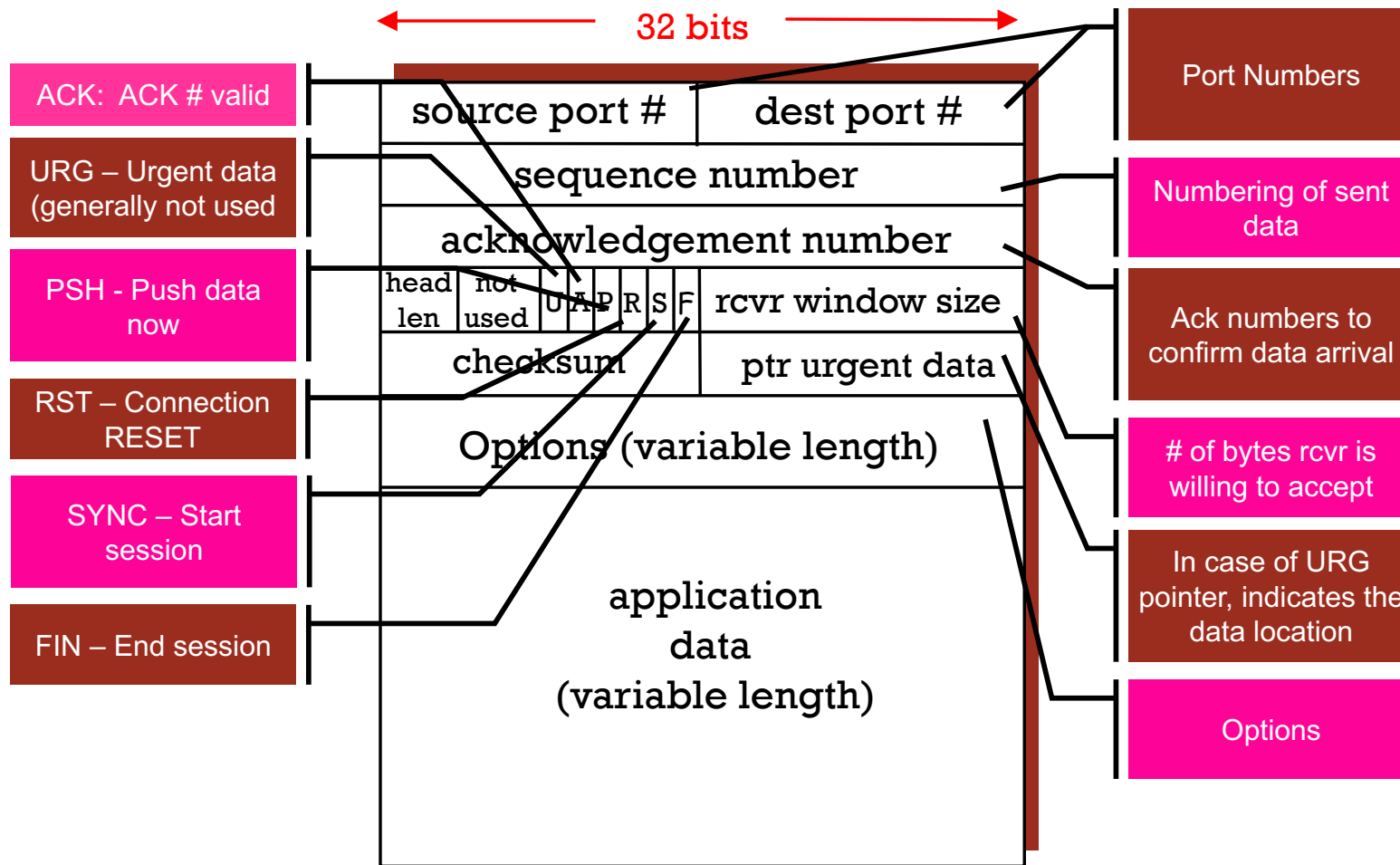
| No. - | Time      | Source        | Destination   | Protocol | Info                                    |
|-------|-----------|---------------|---------------|----------|---|
| 4     | 23.227539 | 1.1.1.1       | 127.0.0.1     | UDP      | Source port: 33333 Destination port: 53 |
| 5     | 23.838867 | 212.179.1.202 | 10.159.3.103  | FTP      | Response: 200 Type set to I.            |
| 6     | 23.857421 | 10.159.3.103  | 212.179.1.202 | FTP      | Request: SIZE upload1_1936              |
| 7     | 23.996093 | 212.179.1.202 | 10.159.3.103  | FTP      | Response: 213 11026917                  |
| 8     | 24.012695 | 10.159.3.103  | 212.179.1.202 | FTP      | Request: MDTM upload1_1936              |
| 9     | 24.208984 | 212.179.1.202 | 10.159.3.103  | FTP      | Response: 213 20071202174050            |
| 10    | 24.266601 | 10.159.3.103  | 212.179.1.202 | FTP      | Request: PASV                           |

Frame 10 (60 bytes on wire, 60 bytes captured)

- Ethernet II, Src: Xerox\_00:00:00 (01:00:01:00:00:00), Dst: d4:c8:20:00:01:00 (d4:c8:20:00:01:00)
- Internet Protocol, Src: 10.159.3.103 (10.159.3.103), Dst: 212.179.1.202 (212.179.1.202)
  - Version: 4
  - Header length: 20 bytes
    - Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
      - 0000 00.. = Differentiated Services Codepoint: Default (0x00)
      - .... ..0. = ECN-Capable Transport (ECT): 0
      - .... ...0 = ECN-CE: 0
  - Total Length: 46
  - Identification: 0x5f49 (24393)
  - Flags: 0x04 (Don't Fragment)
    - 0... = Reserved bit: Not set
    - .1.. = Don't fragment: Set
    - ..0. = More fragments: Not set
  - Fragment offset: 0
  - Time to live: 128
  - Protocol: TCP (0x06)
  - Header checksum: 0xb6fd [correct]
    - [Good: True]
    - [Bad: False]
  - Source: 10.159.3.103 (10.159.3.103)
  - Destination: 212.179.1.202 (212.179.1.202)
- Transmission Control Protocol, Src Port: mps-raft (1700), Dst Port: ftp (21), Seq: 47, Ack: 55, Len: 6
- File Transfer Protocol (FTP)



# ESTRUTURA DA MENSAGEM TCP





# EXEMPLO DO PACOTE TCP

The screenshot displays the Wireshark interface with a packet list and packet details pane. The packet list shows three packets: 9 (FTP Response), 10 (FTP Request), and 11 (FTP Response). Packet 10 is selected, and its details are shown in the packet details pane. The details pane is outlined in red and shows the following information:

- Frame 10 (60 bytes on wire, 60 bytes captured)
- Ethernet II, Src: Xerox\_00:00:00 (01:00:01:00:00:00), Dst: d4:c8:20:00:01:00 (d4:c8:20:00:01:00)
- Internet Protocol, Src: 10.159.3.103 (10.159.3.103), Dst: 212.179.1.202 (212.179.1.202)
- Transmission Control Protocol, Src Port: mps-raft (1700), Dst Port: ftp (21), Seq: 47, Ack: 55, Len: 6
  - Source port: mps-raft (1700)
  - Destination port: ftp (21)
  - [Stream index: 1]
  - Sequence number: 47 (relative sequence number)
  - [Next sequence number: 53 (relative sequence number)]
  - Acknowledgement number: 55 (relative ack number)
  - Header length: 20 bytes
  - Flags: 0x18 (PSH, ACK)
    - 0... .... = Congestion Window Reduced (CWR): Not set
    - .0.. .... = ECN-Echo: Not set
    - ..0. .... = Urgent: Not set
    - ...1 .... = Acknowledgement: Set
    - .... 1... = Push: Set
    - .... .0.. = Reset: Not set
    - .... ..0. = Syn: Not set
    - .... ...0 = Fin: Not set
  - Window size: 16945
  - Checksum: 0x8b8d [validation disabled]
    - [Good Checksum: False]
    - [Bad Checksum: False]
  - [SEQ/ACK analysis]
    - [\[This is an ACK to the segment in frame: 9\]](#)
    - [The RTT to ACK the segment was: 0.057617000 seconds]
    - [Number of bytes in flight: 6]
- File Transfer Protocol (FTP)

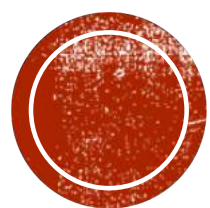


# EXPERIMENTOS COM WIRESHARK

- Abrir wireshark
- Verificar opções de captura
- Primeira captura (acessar site do ITA) e captura a transação HTTP
- Em um terminal, usar o aplicativo ping e capturar
- Investigar Aba Estatísticas
  - Flow Graph (mostra o fluxo de protocolo entre aplicações)
  - Protocol Hierarchy (mostra a hierarquia de encapsulamento)
  - I/O graph
  - Filtrando o TCP Stream da conexão HTTP
    - Explorar TCP stream graphs
- Usar filtros
- Para o lab (ou em casa) analisar os arquivos de exemplo (http, http com jpeg, dns, smtp, dhcp) \*
  - Verifique qual é o endereço MAC e IP dos rastros?
  - Verifique números de sequência do TCP que faixas de números aparecem?
  - Traçar o Flow Graph e investigue os tipos de requisição e resposta e marque-as com cor.

\* <https://wiki.wireshark.org/SampleCaptures>





**MÃOS A OBRA ...**

